**NAPOMENA:** Ovo su odgovori po slajdovima s predavanja. Ti odgovori se mogu proširiti na svakakve načine i nadopuniti s vlastitim znanjem (npr. Razlika između imperativnih i funkcijskih jezika. Svatko ima nešto svoje kao odgovor.), ali ovo je minimum ili, bolje, ono što oni žele čuti.

**Uvod**

1. Što su ekspertni sustavi (ES) i što im je temeljni sadržaj ?

Ekspertni sustavi su programki produkti namijenjeni rješavanju složenih problema u uskoj domeni primjene. Temeljni sadržaj im je predstavljanje i i obradba znanja.

1. Koja četiri pristupa oblikovanju ES razlikujemo ?

Sustav koji djeluje poput ljudi, sustav koji rasuđuje kao ljudi, sustav koji djeluje racionalno, sustav koji rasuđuje racionalno.

1. Što je to racionalni agent ?

Pokazuje nam kako sustav obrađuje podatke koje primi iz okoline i kako reagira na okolinu.

1. Navedi zajedničke značajke i razlike u strukturi predstavljenih racionalnih agenata.

Svi agenti mogu uočiti okolinu i na temelju nekih izbora donositi reakciju. Agent se može dodatno proširiti: 1. s memorijom (koja pamti prošle radnje/stanja/odluke/rezultate); 2. s planiranjem unaprijed, tj. Sustav je u stanju predvidjeti moguća rješenja unaprijed i stoga odabrati najpovoljnije rješenje; 3. S određivanjem korisnosti (kvalitetom rješenja), sustav je u stanju predvidjeti svoja moguća rješenja i ocijeniti kvalitetu tih rješenja i pritom zaključiti koje je rješenje najbolje za sustav (korisnika).

1. Koje su polazne hipoteze u oblikovanju ES utemeljenih na obradi simbola ?

1.Rasuđivanje je obrada simbolima. 2.Svaka obrada simbola može se izvesti na Turingovom stroju.

1. Navedi razine apstrakcija u računalnom sustavu počevši od najniže razine (sklopovlje).

1.Sklopovlje, 2.Strojni jezik, 3.Simbolički strojni jezik (Asemblerska razina), 4.Imperativni programski jezici, 5.Algoritmi i strukture podataka, 6.Razina simbola, 7.Razina znanja

1. Kojoj razini apstrakcija pripada ES implementiran u sustavu xxx (gdje xxx = LISP, sustav logike, sustav s pravilima, …).

LISP – razina simbola, sustav logike – razina znanja, sustav s pravilima – razina znanja

**LISP:**

1. Navedi razliku između imperativnih i funkcijskih programskih jezika.

Imperativni jezici intenzivno koriste pridruživanje, dok funkcijski jezici nemaju pridruživanja, već samo čitaju ulazne vrijednosti i konstruiraju rezultat. Nema promjena ranijih vrijednosti stanja (npr varijabli).

1. Navedi barem 4 značajke LISP-a.

Visoka razina apstrakcije od imperativnih jezika, jer izvođenje ovisi o tipovima parametara (atom, funkcija).

Visoko rekurzivan gleda struktura podataka i algoritama.

Jednakos programa i podataka. Program i podaci su formulirani kao liste.

Implicitan rad s pokazivačima.

Tipovi podataka se ne deklariraju unaprijed, već podaci nose sa sobom tip i kao takvi se evaluiraju.

1. Sintaksa i semantika (petlja izvođenja).

Sintaksa: S-izrazi:atom ili lista. Semantika: Čitaj-evaluiraj-ispiši; evaluiraj=pronađi vrijednost

1. Navedi rezultat izvođenja LISP koda (jednostavan primjer).

Pogledati primjere u ppt, paziti na znak evaluacije '

1. S kojom funkcijom osiguravamo lokalno vezanje simbola.

let

1. Što su to lambda izrazi ?

Prenosi definiciju fukcije izravno u izraz, tj funkcija se ne treba definirati i pridružiti joj se neko ime, već se direktno definira u izrazu.

1. Prikaži navedeni izraz točkastim parovima.

slajdovi

1. Kako se LISP s-izrazi (liste) smještaju u memoriji računala ?

slajdovi

1. Što obuhvaća engleski naziv “garbage collection” ?

Mehanizmi dealokacije memorije (engl. *garbage collection*):

brojanje referencija (sinkrono), "označi i očisti" (asinkrono)

**Logika:**

1. Formalan sustav, definicije temeljnih obilježja.

Definiramo formalan sustav kao dvojku: {Γ, L}

Γ - konačan skup ispravno definiranih formula u odabranom logičkom sustavu

Konzistentan i nekonzistentan (kontradiktoran, npr. P i NE(P))

L – konačan skupa pravila dozvoljenih u odabranom logičkom sustavu

Sustav odrediv (u konačnom vremenu daje odgovor da ili ne), poluodrediv (u konačnom vremenu daje odgovor da, ali za ne ne zna), neodrediv (ako nije ni odrediv ni neodrediv)

Sustav ispravan ako je svaka pravilima dokazana interpretacija ujedno i logička posljedica skupa Γ.

Sustav kompletan ako je svaku logičku posljedicu skupa Γ moguće dokazati pravilima L.

(Za ispravan i kompletan sustav najbolje provjeriti tablicu na slajdu 32)

...

1. Primjeri nekih obilježja.

Propozicijska logika je ispravno, kompletna i odrediva.

Predikatna logika je poluodrediva. Ispravna i kompletna za „jednostavnu“ predikatnu logiku, bez nekih varijabli s problematičnim skupom (npr aritmetika).

1. Preslikavanje izjavnih rečenica prirodnog jezika u dobro definirane formule predikatne logike.

Just do it.

1. Normalni oblici logičkih formula.

Svaka propozicijska formula se može pretvoriti u jedan od ova dva oblika:

Disjunkcijski normalni oblik (DNF) (k1 AND k2 AND k3...) OR (p1 AND p2...)

Konjukcijski normalni oblik (CNF) (k1 OR k2 OR k3...) AND (p1 OR p2...)

1. Sat problem.

Problem zadovoljivosti je NP problem (3 ili više literala). Iscrpnim pretraživanjem nećemo nigjde stići. Sva sreća stohastičkim algoritmima (tj oni koji se temelje uglavnom na vjerojatnostima) se uspjeva dobiti rješenje u polinomnom vremenu.

Dodatno: to su algoritmi koji se temelje na rješavanju sukoba (ako nađu sukob mogu eliminirati neke buduće pretrage) i algoritmi lokalnog pretraživanja (stohastički).

1. Teorem dedukcije.

Formula A je logička posljedica formule B ako je (A=>B) tautologija.

DeMorgan nam je rekao kako je (A=>B) isto kao (NOT (A) AND B) i s time spasio stvar i olakšao nam zaključivanje o tautologiji, jer umjesto da ispitujemo svaku moguću interpretaciju te formule, možemo uzeti negaciju te formule (A AND NOT(B)) i pridružiti je ostalim formulama u skupu i vidjeti hoćemo li dobiti prazan skup, nakon primjene pravila. Rezultat nam je da nismo isprobavali puuno T i F-ova, nego samo križali mali broj formula.

Genijalno.

1. Pravilo razrješavanja (Robinson) i svođenje elementarnih pravila na razrješavanje.

Logička posljedica dviju istinitih, konjunkcijom vezanih, univerzalno kvantificiranih normaliziranih klauzula (pazi sve formule moraju biti normalizirane klauzule) je normalizirana klauzula bez jednog komplementarnog para literala.

Teška definicjia; uglavnom: Ako uzmemo dvije klauzule koje imaju komplementarni literal, možemo ih spojiti i dobiti jednu klauzulu koja je „povezana“ a bez tog literala. Zapravo smo napravili nešto kao supstituciju, tj spojili dvije klauzule preko jednog literala kojeg smo izbacili.

1. Normalizirane klauzule.

Formula oblika disjunkcije literala.

1. Izjednačavanje atomičkih predikata.

Valjda primjer, razrješavanja preko izjednačavanja (gore opisana supstitucija).

1. Navedi algoritam za opći postupak dokazivanja teorema.

1.Definiraj skup aksioma E kao ispravne formule (wff).

2.Definiraj teorem H kao ispravnu formulu (wff).

3.Negirani H dodaj skupu aksioma E, dobiven je skup formula F.

4.Preslikaj skup formula F u skup univerzalno kvantificiranih normaliziranh klauzula K.

Izaberi dvije klauzuje i razrješavaj ih, dok se ne dobije () prazni skup (teorem dokazan) ili redukcija više nije moguća (teorem nije dokazan).

1. Strategije u razrješavanju.

1.Strategija izbora u širinu – uzme se jedna klauzula i usporedi se sa svima, pa druga se usporedi sa svima itd do kraja.

2.Strategija s prvenstvom jediničnih klauzula – Uzme se najmanja klauzula i nad njom se pokušava dobiti () prazan skup, jer bi ona potencijalno mogla najbrže dovoditi do praznog skupa kada ima najmanji broj literala.

3.Strategija skupa potpore – uglavnom pretražuje manje skupove, a ne potpune, jer gleda jel ima koji potomak i na temelju nekih pravila pokušava zaključiti je li skup potpore kompletan (je li za svaku logičku posljedicu moguće dokazati pravilima).

4.Linearna ulazna strategija razrješavanja – Uzima jednu po jednu klauzulu (na početku dvije) i prenosi rezultat tako da ga uspoređuje sa slijedećom klauzulom. Nije kompletna zbog toga što može naletjeti na ciklus koji umjesto da se poništi (prazni skup, kontradikcija) on se zavrti s prenošenjem jednog literala.

1. Pojednostavljivanje u skupu formula predikatne logike.

1.Uporaba temeljnih pravila npr (A AND B AND A) jedan A očito ne treba (A AND B)

2.Podrazumijevanje npr (A AND B AND C) onda se podrazumijeva i podskup (A AND B)

1. Goedelov teorem nekompletnosti.

Postoje izrazi koji su istiniti ali nedokazljivi ili dokazljivi ali ne istiniti.

Kada se u FOPL ubaci aritmetika i skup prirodnih brojeva N.

(Ovo je dovoljno znati i Godelu, sve više je previše.)